

LAPORAN MATA KULIAH PEMROGRAMAN MOBILE

Dosen Pengajar : Ade Ismail, S.Kom., M.TI.

Jobsheet - 4



Nama : Surya Rahmat Fatahillah

NIM : 2341760020

Prodi : Sistem Informasi Bisnis

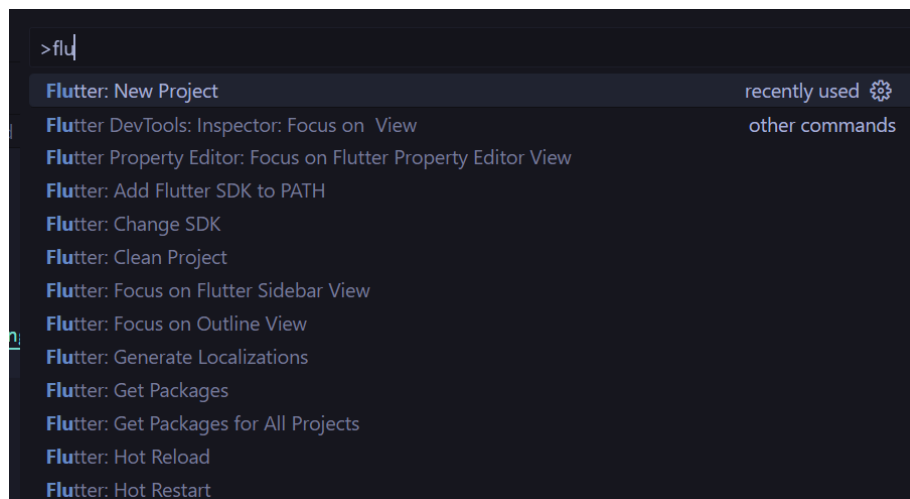
Kelas: SIB 3D

**JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG 2025**

Praktikum 1: Membuat Project Flutter Baru

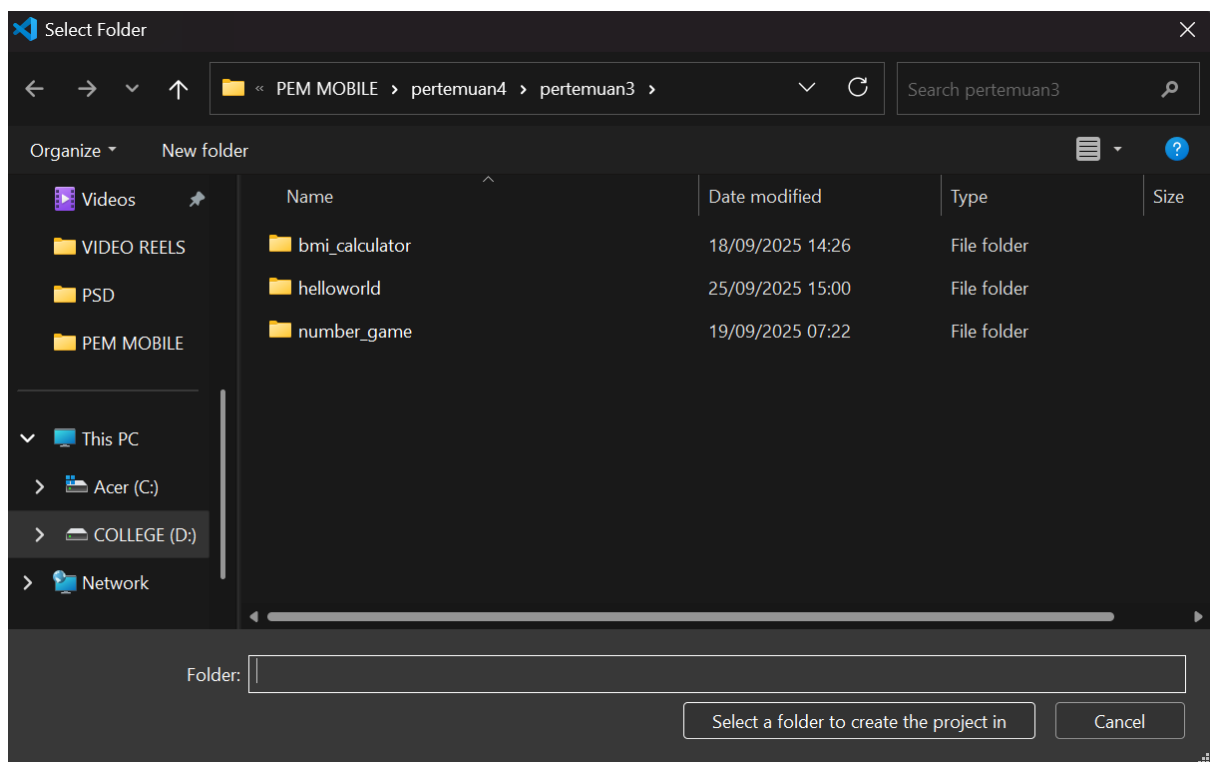
Langkah 1:

Buka VS Code, lalu tekan tombol **Ctrl + Shift + P** maka akan tampil *Command Palette*, lalu ketik **Flutter**. Pilih **New Application Project**.



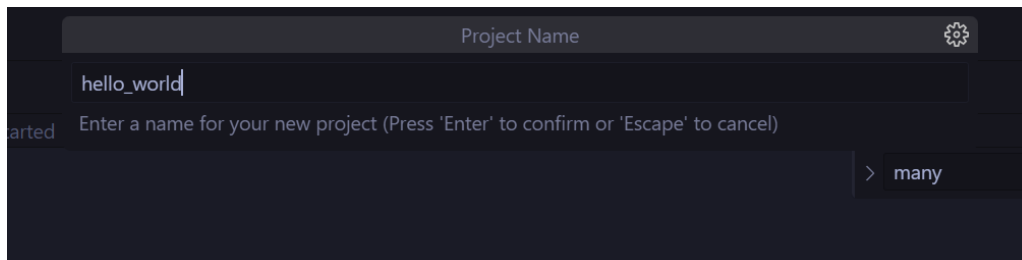
Langkah 2:

Kemudian buat folder sesuai style laporan praktikum yang Anda pilih. Disarankan pada folder dokumen atau desktop atau alamat folder lain yang tidak terlalu dalam atau panjang. Lalu pilih **Select a folder to create the project in**.



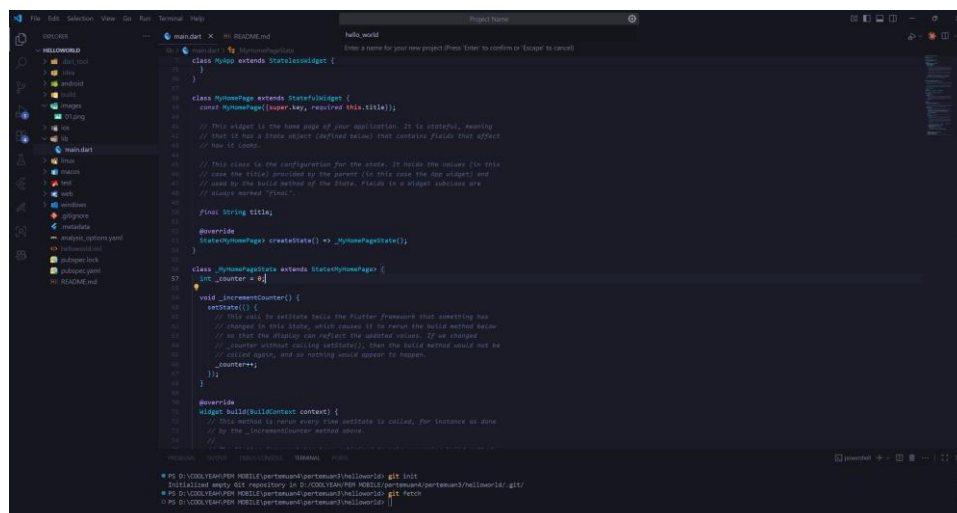
Langkah 3:

Buat nama project flutter **hello_world** seperti berikut, lalu tekan **Enter**. Tunggu hingga proses pembuatan project baru selesai.



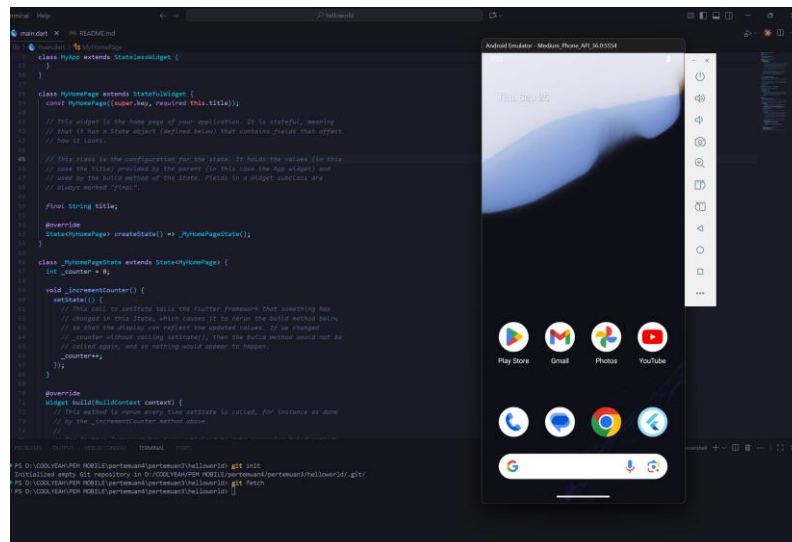
Langkah 4:

Jika sudah selesai proses pembuatan project baru, pastikan tampilan seperti berikut. Pesan akan tampil berupa **"Your Flutter Project is ready!"** artinya Anda telah berhasil membuat project Flutter baru.



Praktikum 2: Menghubungkan Perangkat Android atau Emulator

Melanjutkan dari praktikum 1, Anda diminta untuk menjalankan aplikasi ke perangkat fisik (device Android atau iOS). Silakan ikuti langkah-langkah pada codelab tautan berikut ini.



Praktikum 3: Membuat Repository GitHub dan Laporan Praktikum

Langkah 1:

Login ke akun [GitHub](#) Anda, lalu buat repository baru dengan nama "**flutter-fundamental-part1**"

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

1 General

Owner *

SuryaRf

Repository name *

flutter-fundamental-part1

Your new repository will be created as flutter-fundamental-part1-.

The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. How about [legendary-guacamole](#)?

Description

#16 | Flutter Fundamental - Bagian 1 (TSA 2022)

47 / 350 characters

2 Configuration

Choose visibility *

Public

Choose who can see and commit to this repository

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

Off

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

3 Jumpstart your project with Copilot (optional)

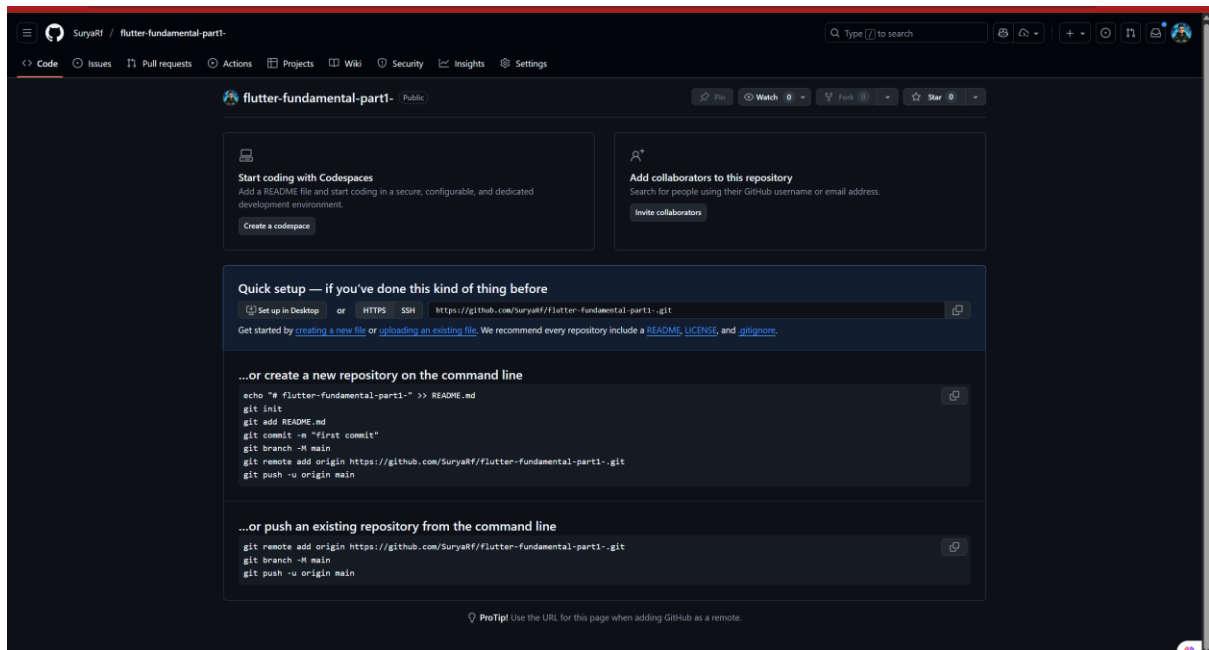
Tell Copilot what you want to build in this repository. After creation, Copilot will open a pull request with generated files - such as a basic app, starter code, or other features you describe - then request your review when it's ready. [About Copilot coding agent](#)

Prompt

Describe what you want Copilot to build

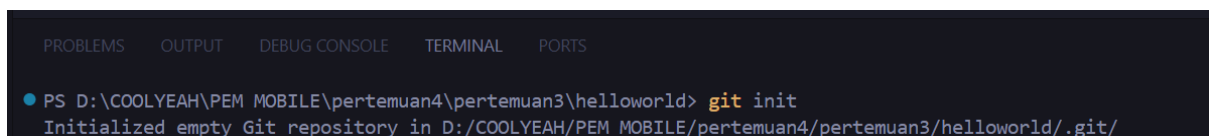
Langkah 2:

Lalu klik tombol "**Create repository**" lalu akan tampil seperti gambar berikut.



Langkah 3:

Kembali ke VS code, project flutter hello_world, buka terminal pada menu **Terminal > New Terminal**. Lalu ketik perintah berikut untuk inisialisasi git pada project Anda.



Langkah 4:

Pilih menu **Source Control** di bagian kiri, lalu lakukan **stages (+)** pada file **.gitignore** untuk mengunggah file pertama ke repository GitHub.

Langkah 5:

Beri pesan commit "**tambah gitignore**" lalu klik **Commit (✓)**

Langkah 6:

Lakukan push dengan klik bagian menu titik tiga > **Push**

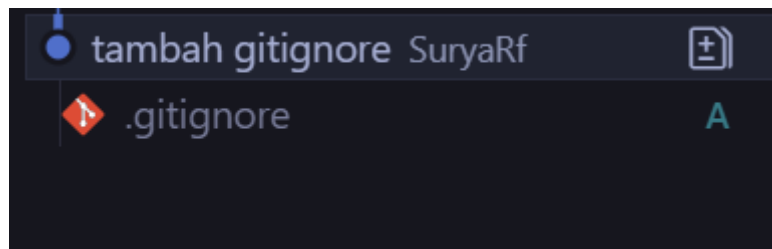
Langkah 7:

Di pojok kanan bawah akan tampil seperti gambar berikut. Klik "**Add Remote**"

Langkah 8:

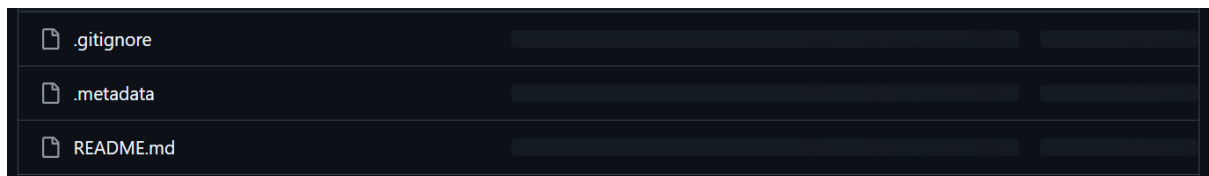
Salin tautan repository Anda dari browser ke bagian ini, lalu klik **Add remote**

Setelah berhasil, tulis remote name dengan "**origin**"



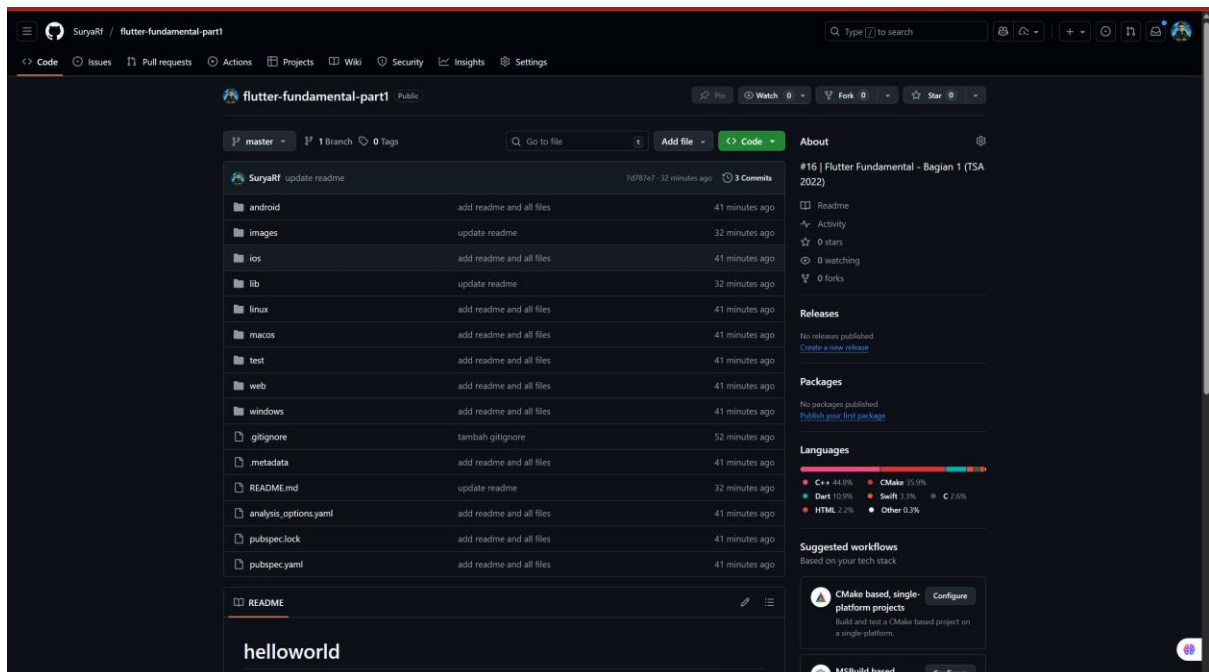
Langkah 9:

Lakukan hal yang sama pada file **README.md** mulai dari Langkah 4. Setelah berhasil melakukan push, masukkan username GitHub Anda dan password berupa token yang telah dibuat (pengganti password konvensional ketika Anda login di browser GitHub). Reload halaman repository GitHub Anda, maka akan tampil hasil push kedua file tersebut seperti gambar berikut.



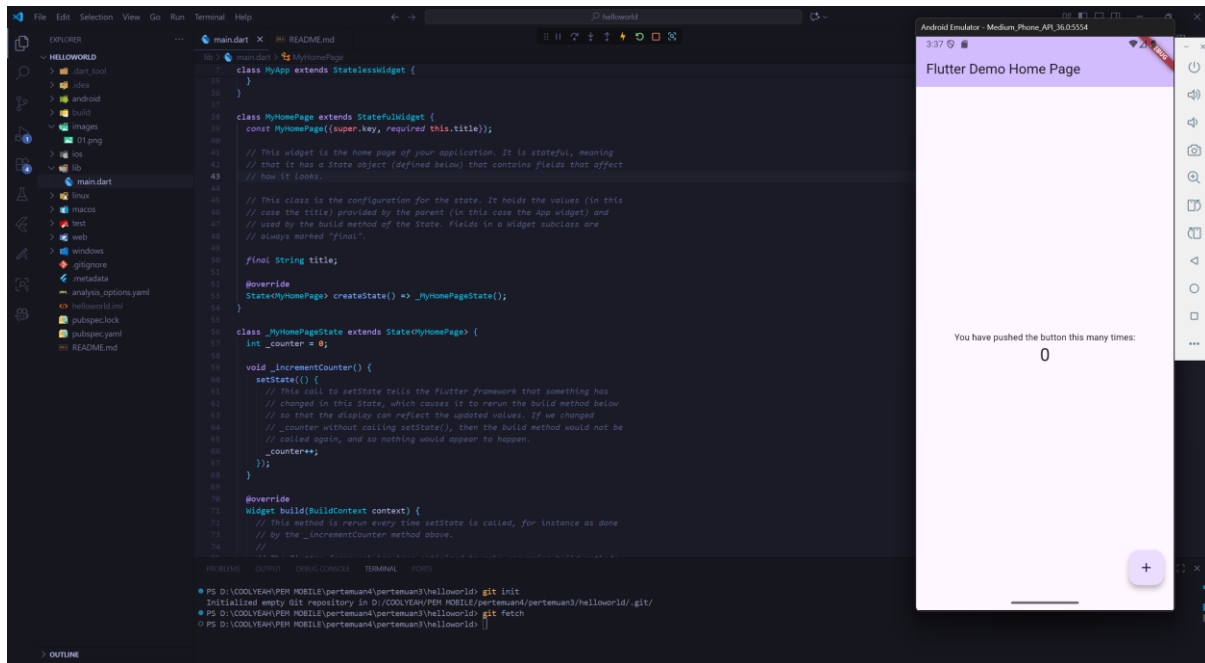
Langkah 10:

Lakukan push juga untuk semua file lainnya dengan pilih **Stage All Changes**. Beri pesan commit "**project hello_world**". Maka akan tampil di repository GitHub Anda seperti berikut.



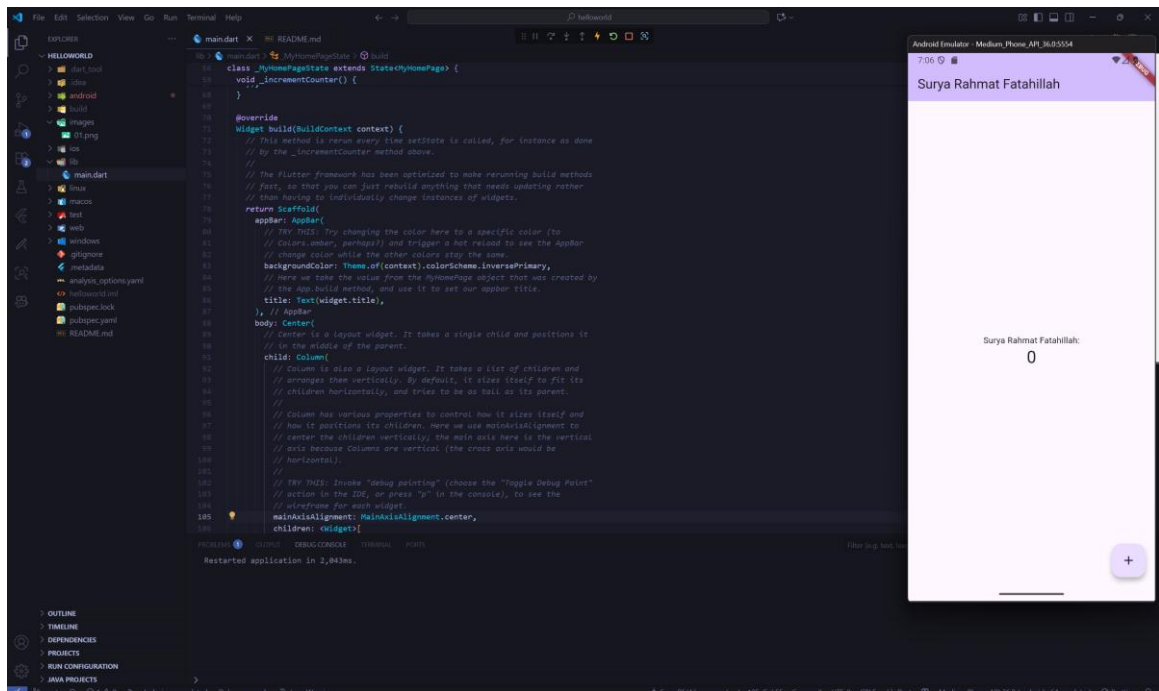
Langkah 11:

Kembali ke VS Code, ubah platform di pojok kanan bawah ke emulator atau device atau bisa juga menggunakan browser Chrome. Lalu coba running project **hello_world** dengan tekan **F5** atau **Run > Start Debugging**. Tunggu proses kompilasi hingga selesai, maka aplikasi flutter pertama Anda akan tampil seperti berikut.



Langkah 12:

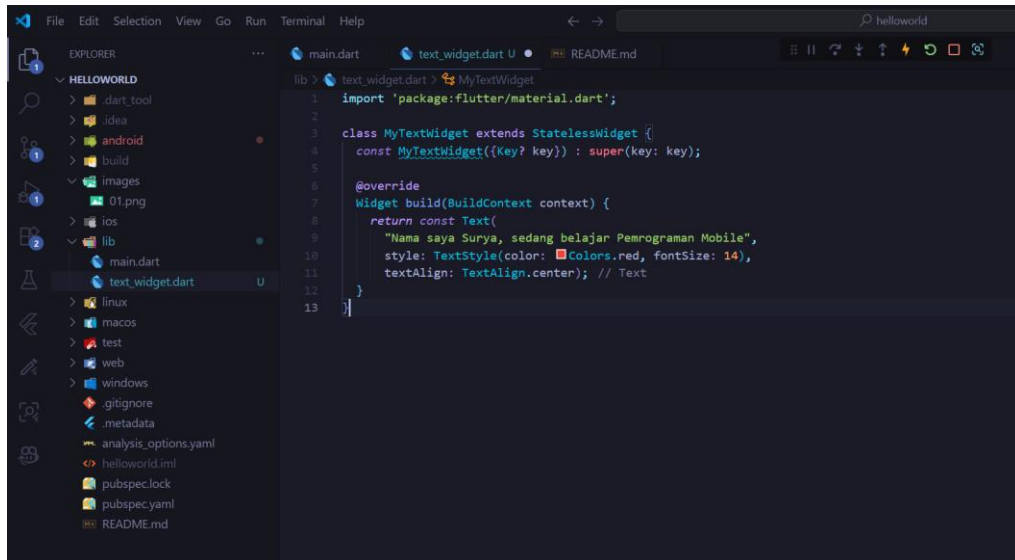
Silakan screenshot seperti pada Langkah 11, namun teks yang ditampilkan dalam aplikasi berupa nama lengkap Anda. Simpan file screenshot dengan nama **01.png** pada folder **images** (buat folder baru jika belum ada) di project **hello_world** Anda. Lalu ubah isi README.md seperti berikut, sehingga tampil hasil screenshot pada file README.md. Kemudian push ke repository Anda.



Praktikum 4: Menerapkan Widget Dasar

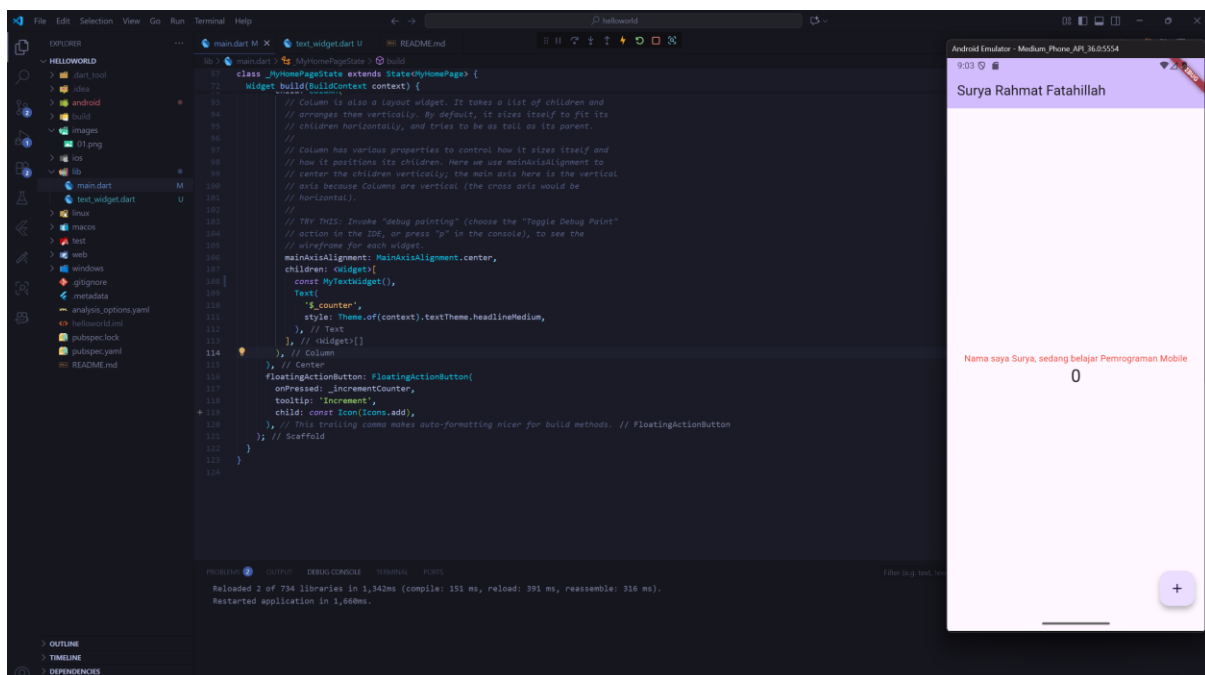
Langkah 1: Text Widget

Buat folder baru **basic_widgets** di dalam folder **lib**. Kemudian buat file baru di dalam **basic_widgets** dengan nama **text_widget.dart**. Ketik atau salin kode program berikut ke project **hello_world** Anda pada file **text_widget.dart**.



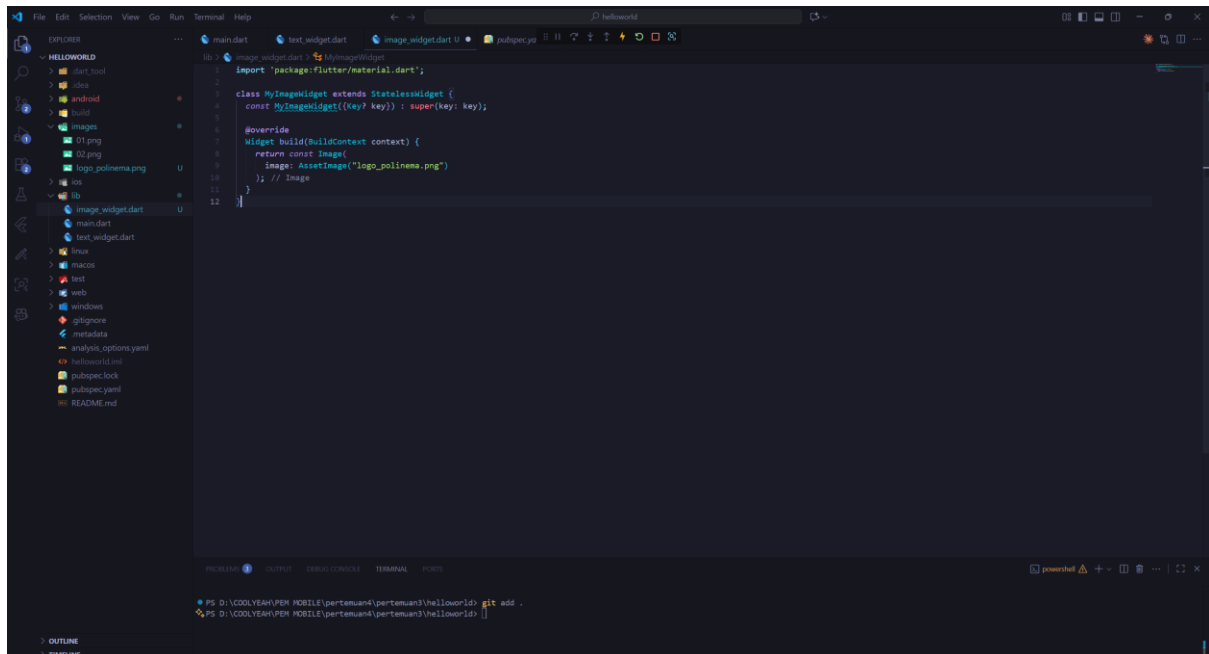
```
File Edit Selection View Go Run Terminal Help
lib > text_widget.dart > MyTextWidget
1 import 'package:flutter/material.dart';
2
3 class MyTextWidget extends StatelessWidget {
4   const MyTextWidget({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return const Text(
9       "Nama saya Surya, sedang belajar Pemrograman Mobile",
10      style: TextStyle(color: Colors.red, fontSize: 14),
11      textAlign: TextAlign.center); // Text
12   }
13 }
```

Lakukan import file **text_widget.dart** ke **main.dart**, lalu ganti bagian **text widget** dengan kode di atas. Maka hasilnya seperti gambar berikut. Screenshot hasil milik Anda, lalu dibuat laporan pada file **README.md**.

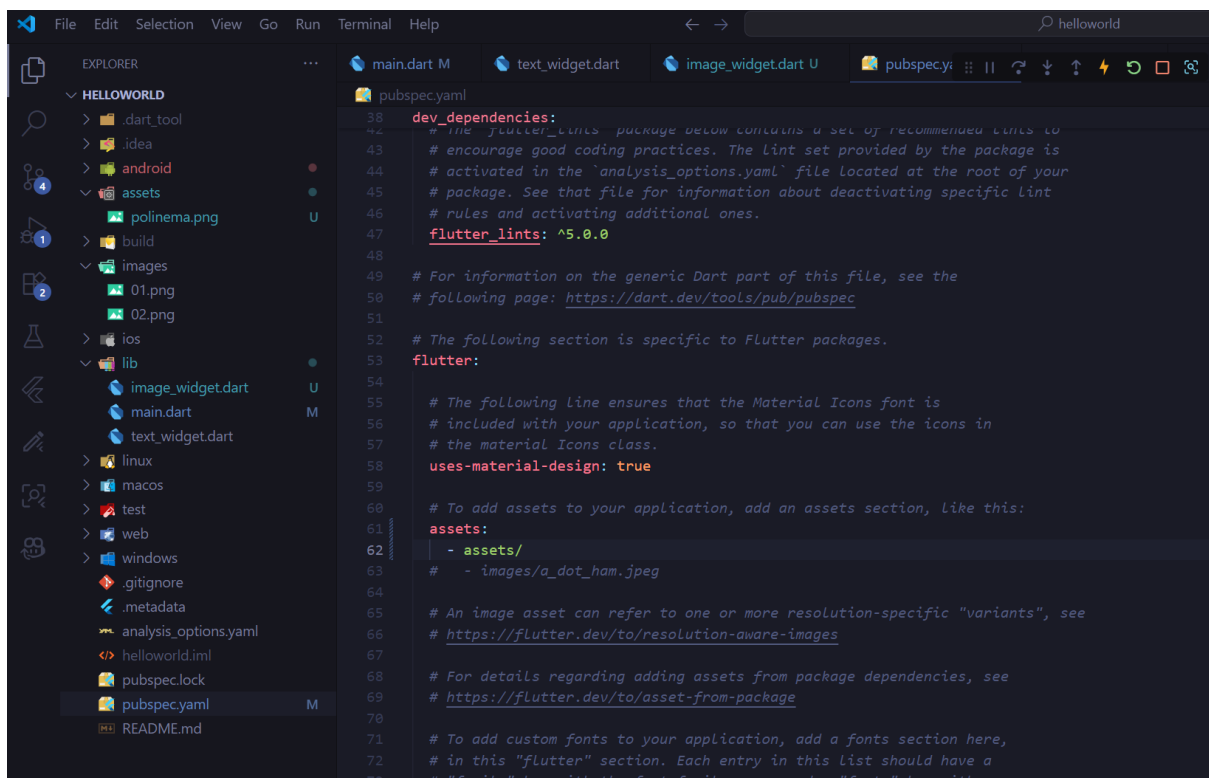


Langkah 2: Image Widget

Buat sebuah file `image_widget.dart` di dalam folder `basic_widgets` dengan isi kode berikut.



Lakukan penyesuaian asset pada file `pubspec.yaml` dan tambahkan file logo Anda di folder `assets` project `hello_world`.

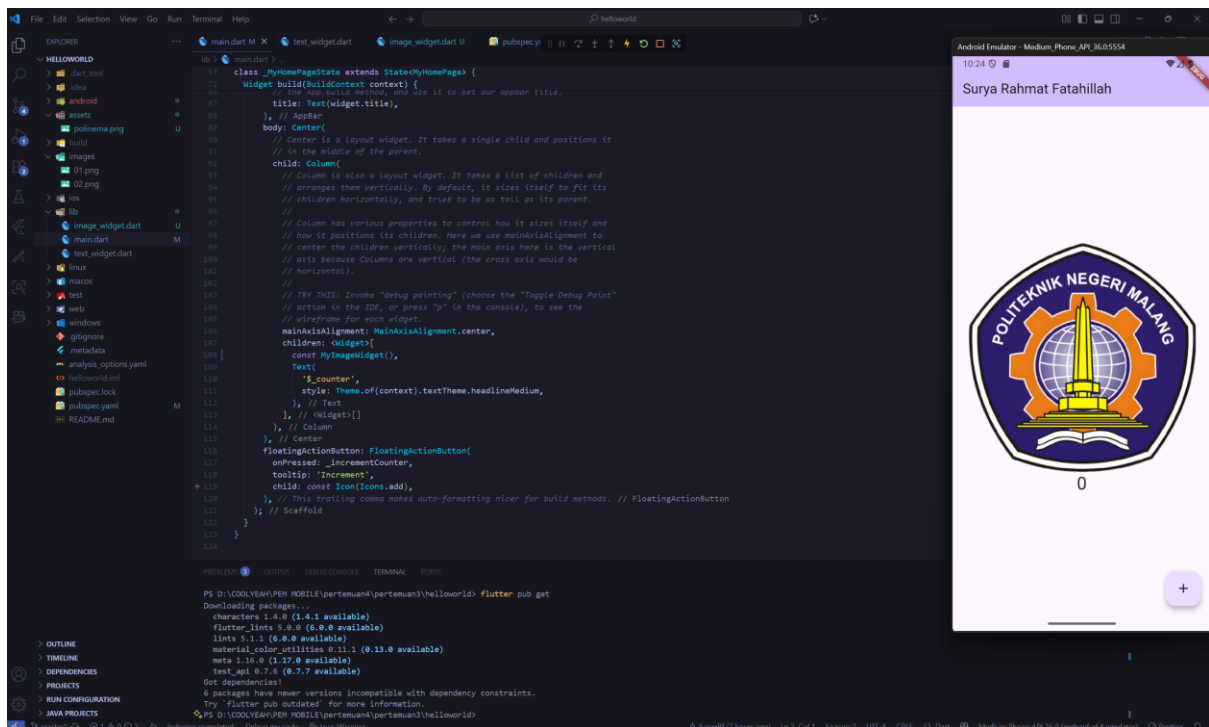


Jangan lupa sesuaikan kode dan import di file `main.dart` kemudian akan tampil gambar seperti berikut.

```

84 // action in the IDE, or press "p" in the console), to see the
85 // wireframe for each widget.
86 mainAxisAlignment: MainAxisAlignment.center,
87 children: <Widget>[
88   const MyImageWidget(),
89   Text(
90     '$_counter',
91     style: Theme.of(context).textTheme.headlineMedium,
92   ), // Text
93 ], // <Widget>[]
94 ), // Column
95 ), // Center
96 floatingActionButton: FloatingActionButton(
97   onPressed: _incrementCounter,
98   tooltip: 'Increment',
99   child: const Icon(Icons.add),
100 ), // This trailing comma makes auto-formatting nicer for build methods. // FloatingActionButton
101 ); // Scaffold
102 }
103 }
104

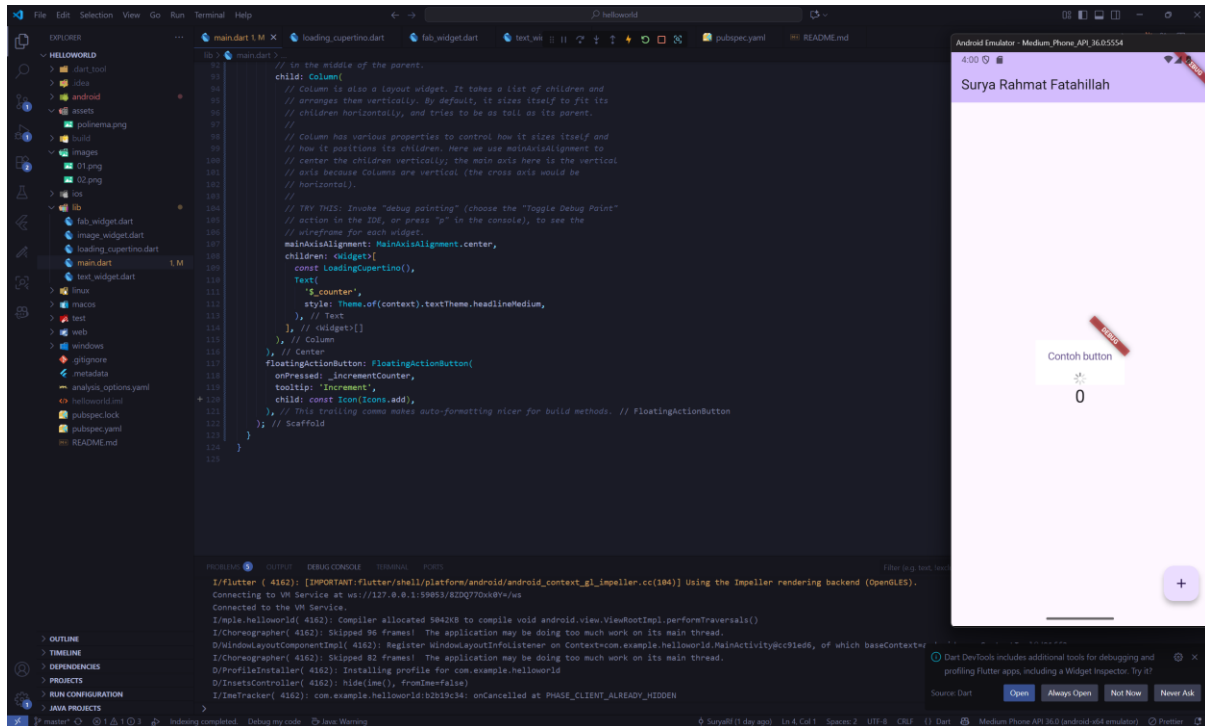
```



Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

Langkah 1: Cupertino Button dan Loading Bar

Buat file di `basic_widgets > loading_cupertino.dart`. Import stateless widget dari material dan cupertino. Lalu isi kode di dalam method `Widget build` adalah sebagai berikut.

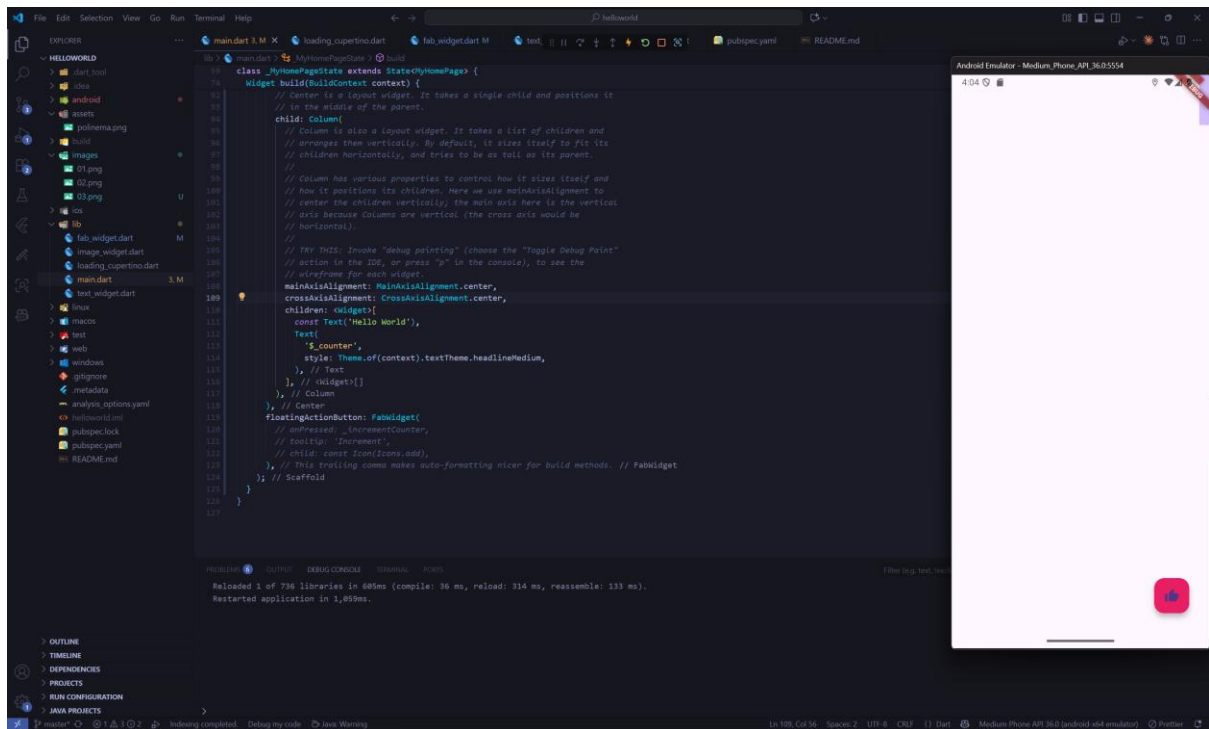


Langkah 2: Floating Action Button (FAB)

Button widget terdapat beberapa macam pada flutter yaitu `AppBar`, `DropDownButton`, `TextButton`, `FloatingActionButton`, `IconButton`, `OutlineButton`, `PopupMenuButton`, dan `ElevatedButton`.

Buat file di `basic_widgets > fab_widget.dart`. Import stateless widget dari material. Lalu isi kode di dalam method `Widget build` adalah sebagai berikut.

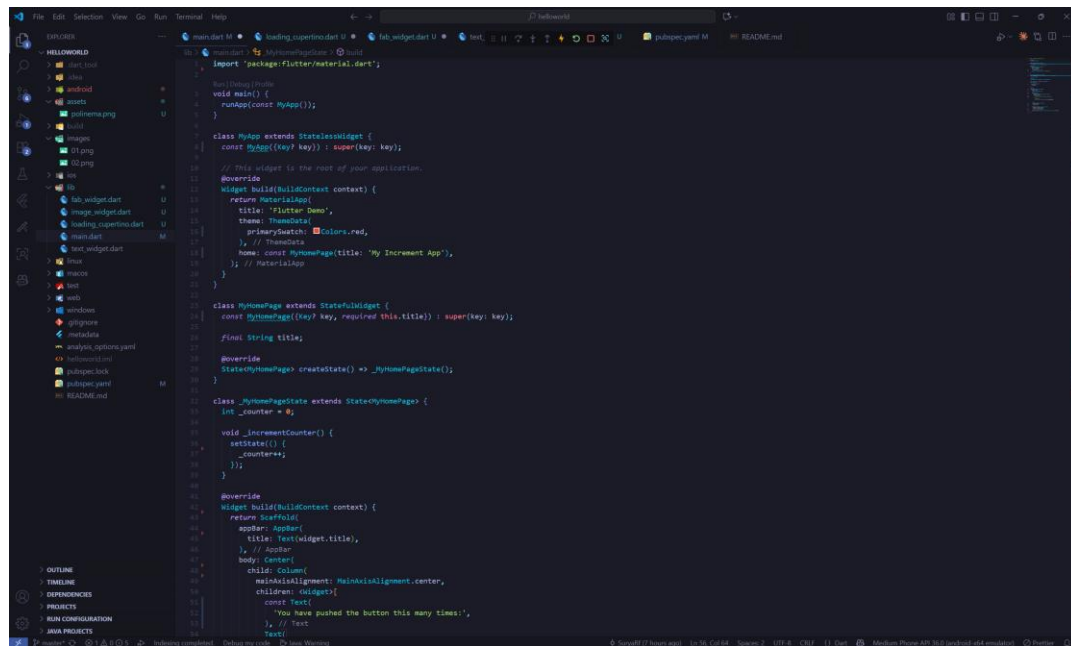
```
lib > fab_widget.dart > FabWidget
1 import 'package:flutter/material.dart';
2
3 class FabWidget extends StatelessWidget {
4   const FabWidget({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return MaterialApp(
9       home: Scaffold(
10        floatingActionButton: FloatingActionButton(
11          onPressed: () {
12            // Add your onPressed code here!
13          },
14          child: const Icon(Icons.thumb_up),
15          backgroundColor: Colors.pink,
16        ), // FloatingActionButton
17      ), // Scaffold
18    ); // MaterialApp
19  }
20
21 }
```



Langkah 3: Scaffold Widget

Scaffold widget digunakan untuk mengatur tata letak sesuai dengan material design.

Ubah isi kode main.dart seperti berikut.



```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.red,
      ),
      home: const MyHomePage(title: 'My Increment App'),
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({key? key, required this.title}) : super(key: key);

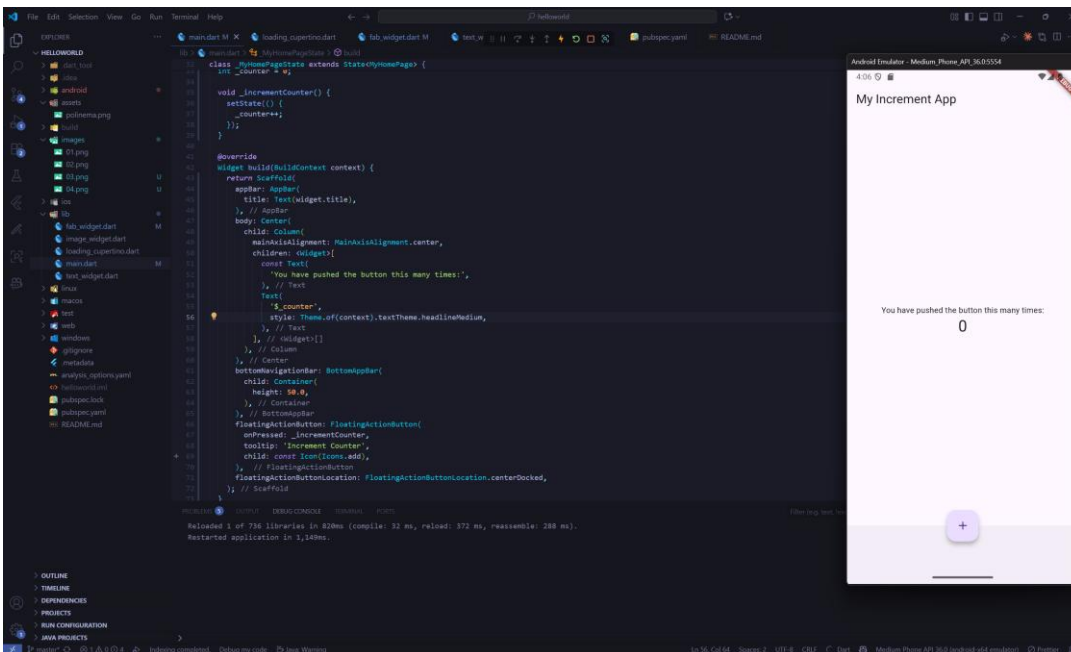
  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment Counter',
        child: const Icon(Icons.add),
      ), // FloatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
    ); // Scaffold
  }
}
```



```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.red,
      ),
      home: const MyHomePage(title: 'My Increment App'),
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({key? key, required this.title}) : super(key: key);

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

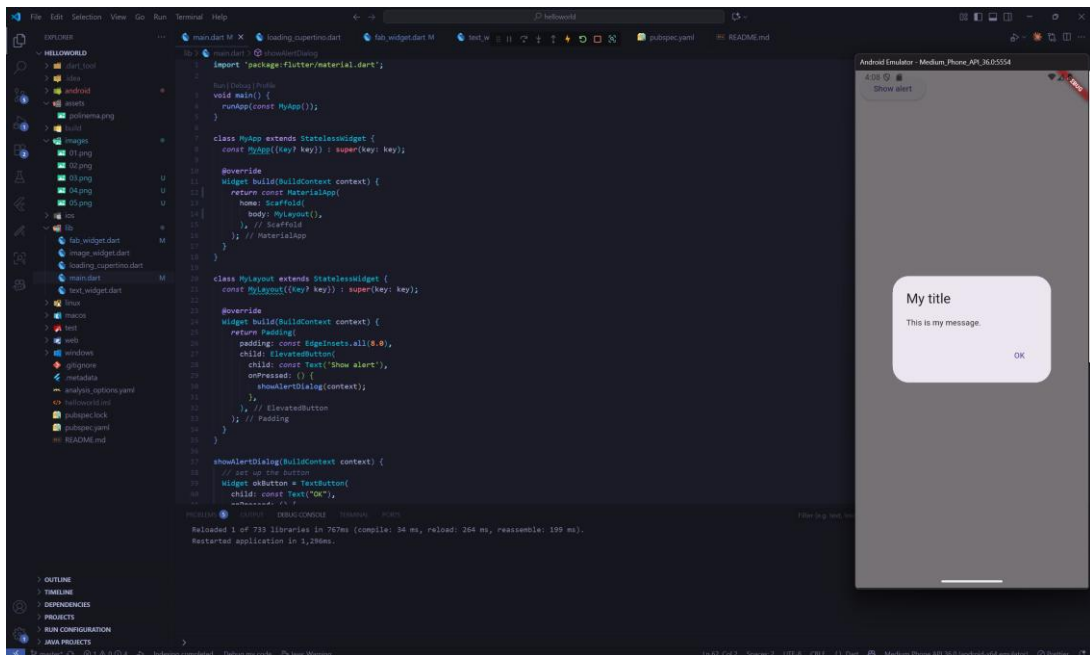
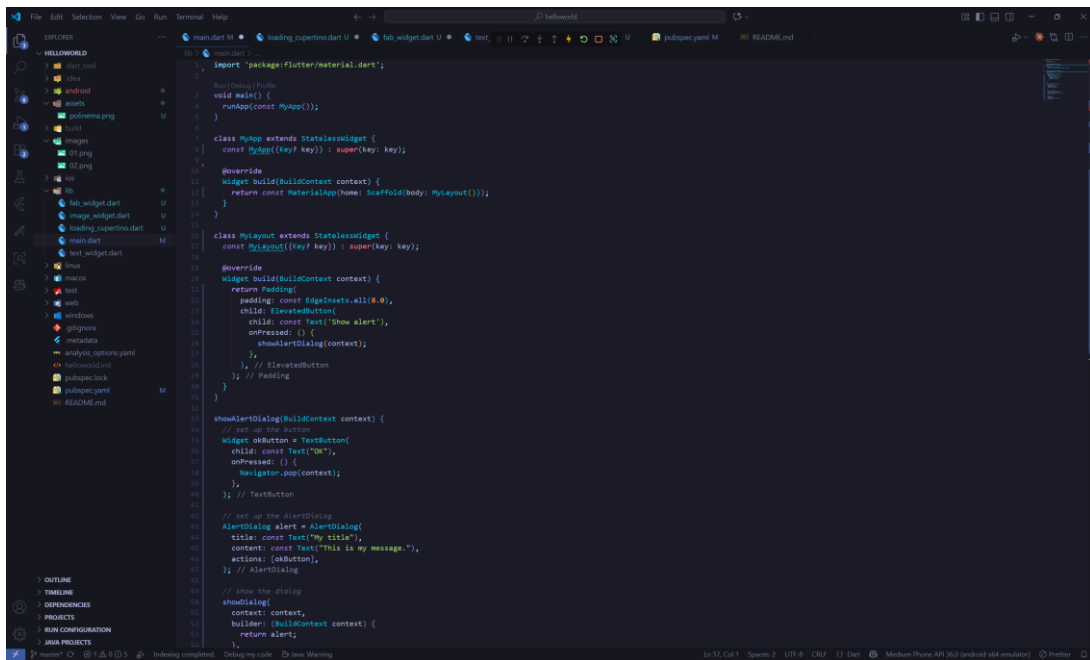
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment Counter',
        child: const Icon(Icons.add),
      ), // FloatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
    ); // Scaffold
  }
}
```

Langkah 4: Dialog Widget

Dialog widget pada flutter memiliki dua jenis dialog yaitu AlertDialog dan SimpleDialog.

Ubah isi kode main.dart seperti berikut.



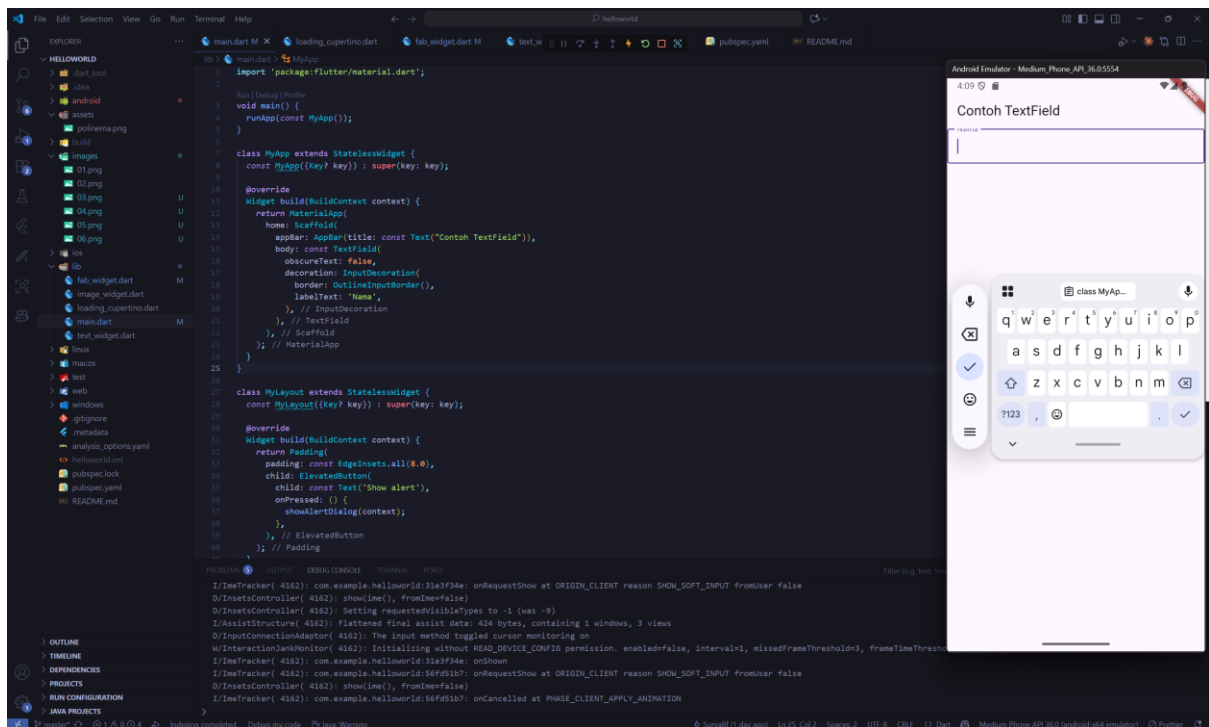
Langkah 5: Input dan Selection Widget

Flutter menyediakan widget yang dapat menerima input dari pengguna aplikasi yaitu antara lain Checkbox, Date and Time Pickers, Radio Button, Slider, Switch, TextField.

Contoh penggunaan TextField widget adalah sebagai berikut:

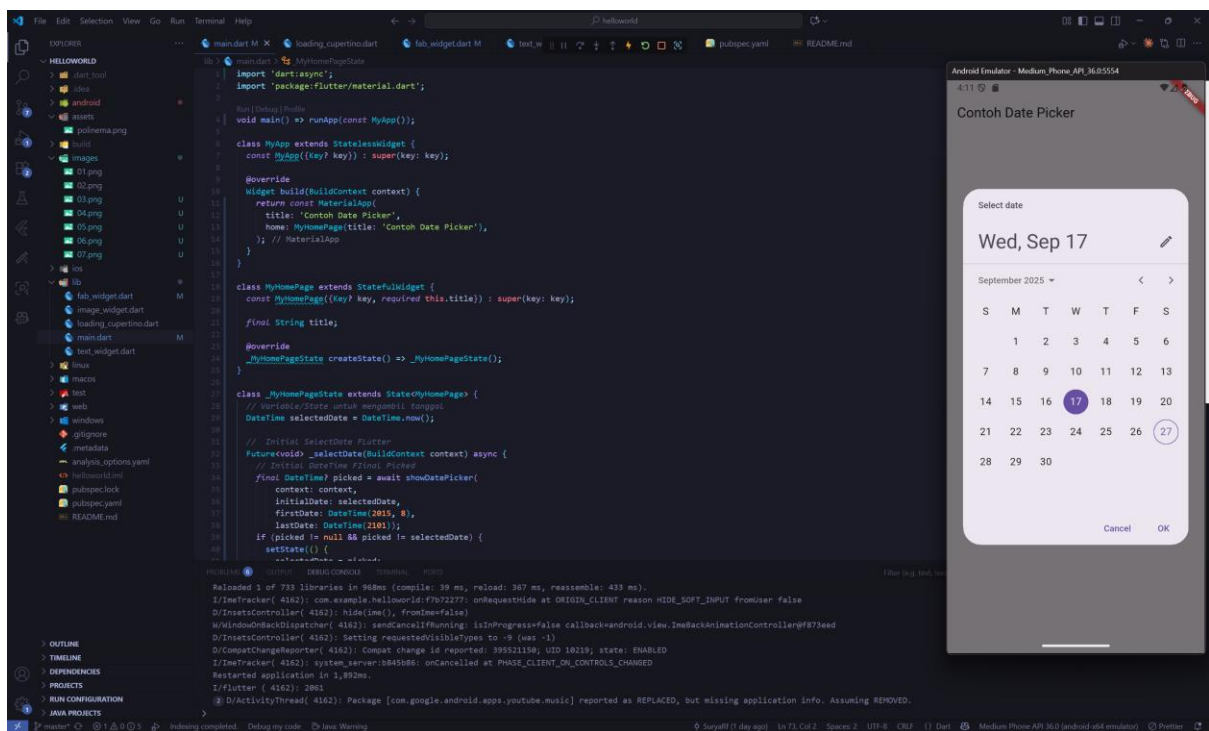
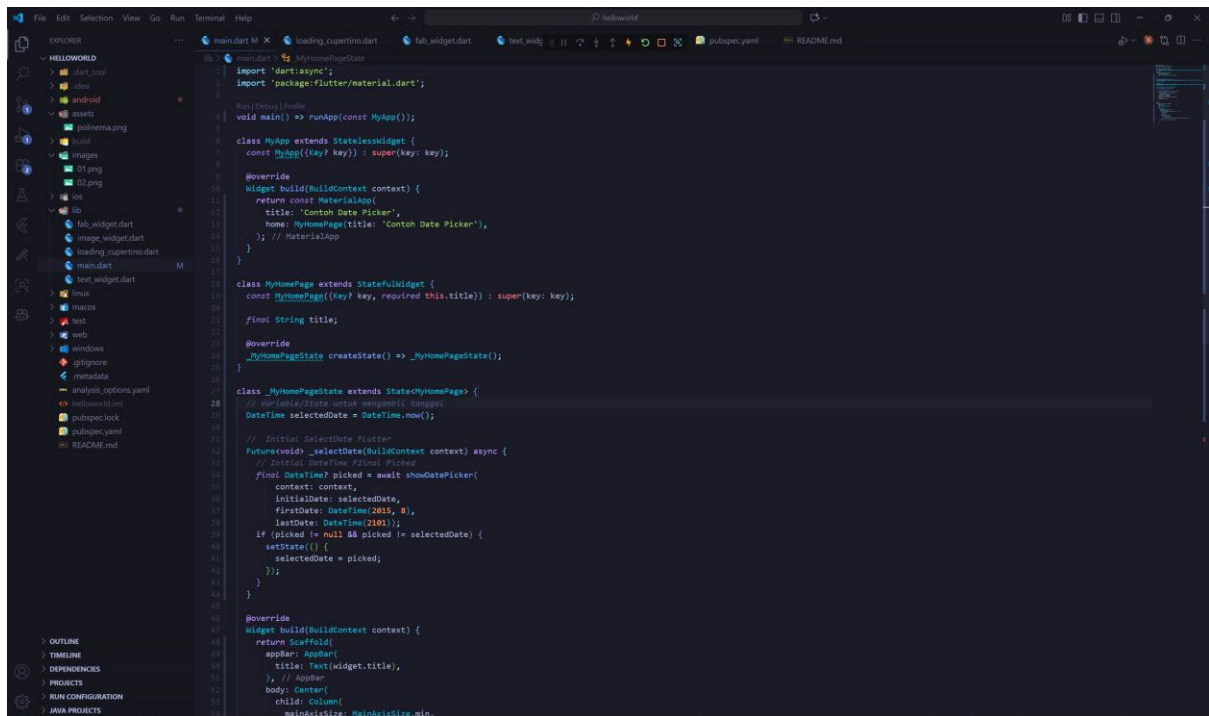
```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Contoh TextField")),
        body: const TextField(
          obscureText: false,
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: 'Nama',
          ), // InputDecoration
        ), // TextField
      ), // Scaffold
    ); // MaterialApp
  }
}
```



Langkah 6: Date and Time Pickers

Date and Time Pickers termasuk pada kategori input dan selection widget, berikut adalah contoh penggunaan Date and Time Pickers.



Github:

<https://github.com/SuryaRf/flutter-fundamental-part1>